Fig. 1, Prior Art

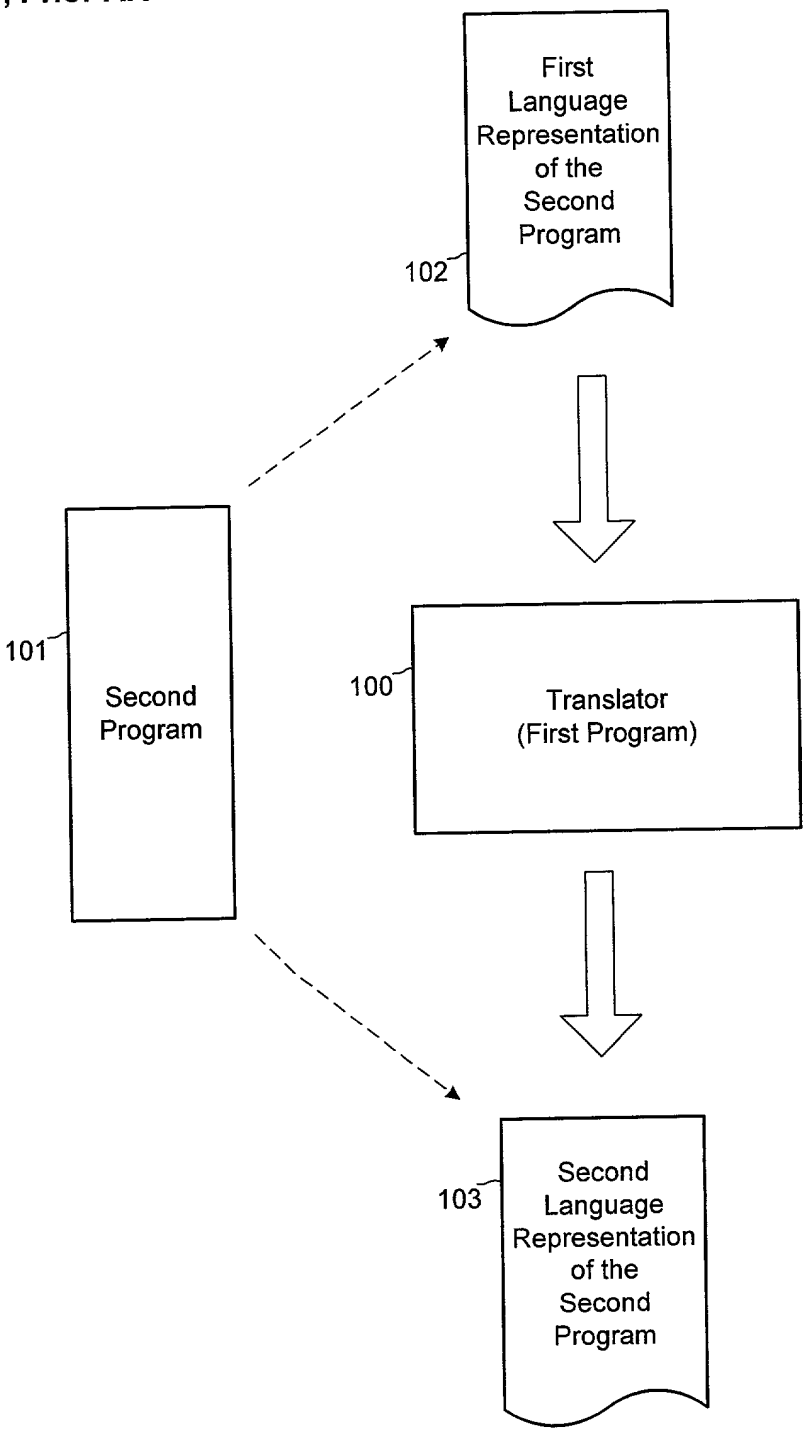


Fig. 2, Prior Art

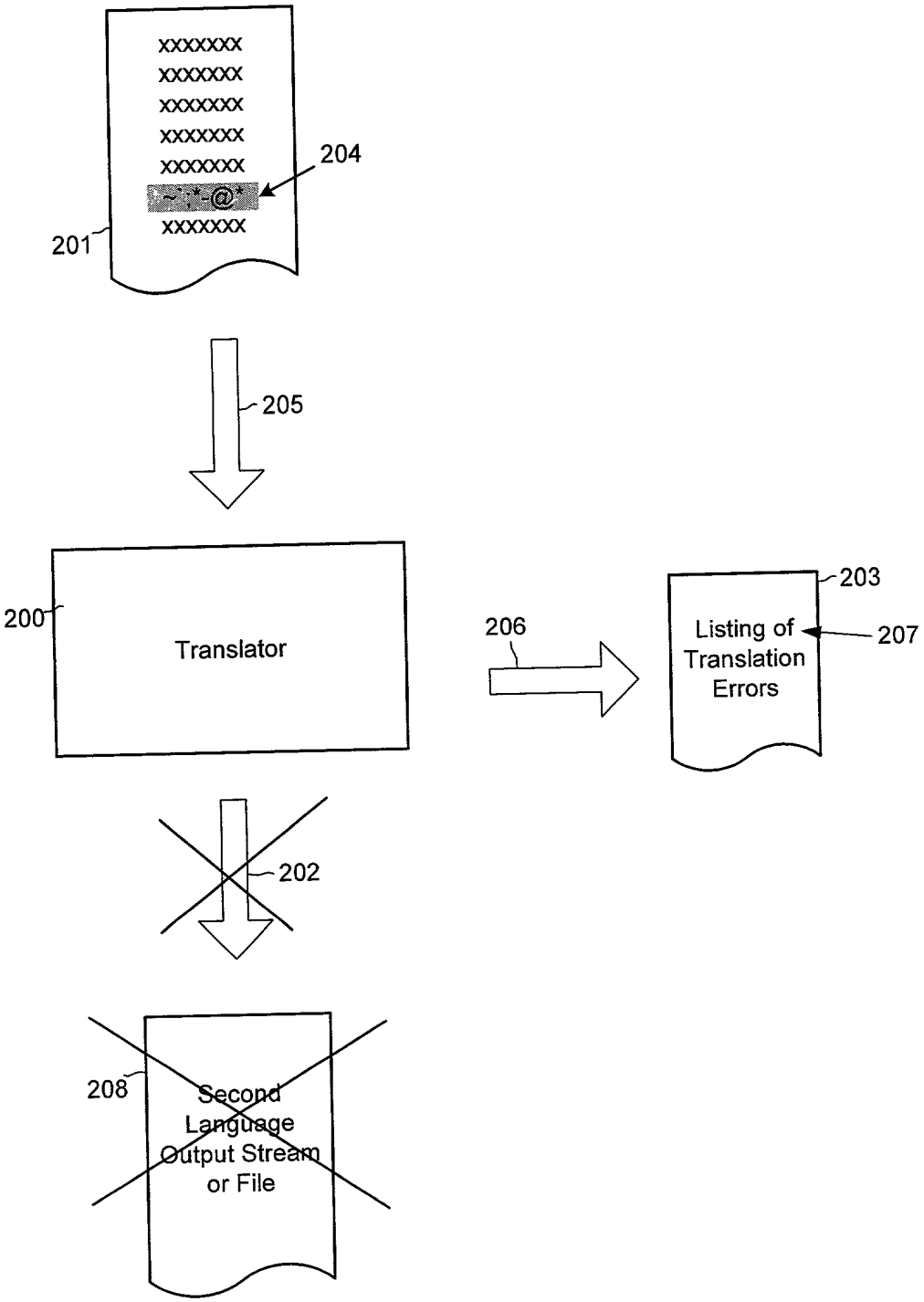


Fig. 3

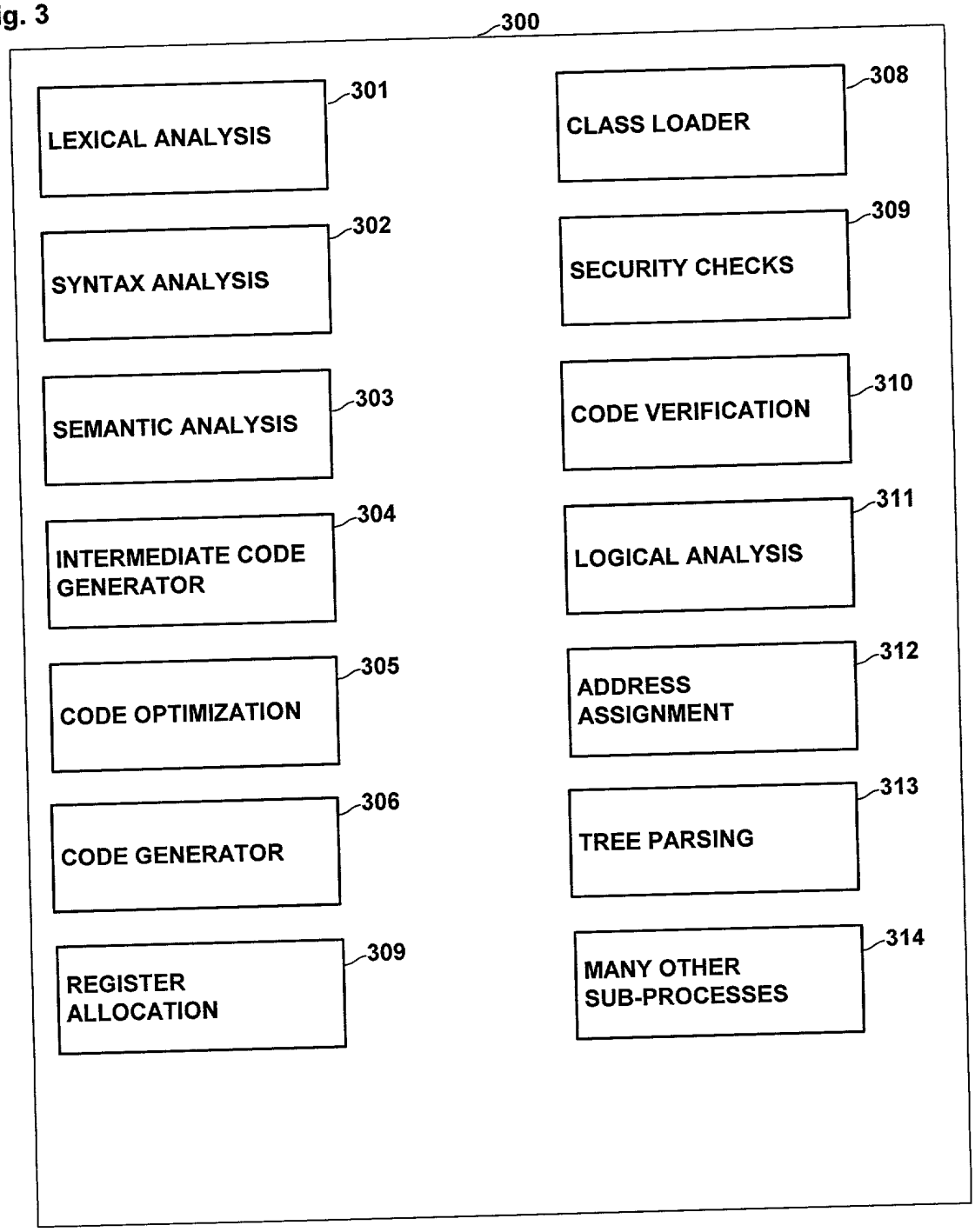

300
LEXICAL ANALYSIS
301
SYNTAX ANALYSIS
302
SEMANTIC ANALYSIS
303
INTERMEDIATE CODE GENERATOR
304
CODE OPTIMIZATION
305
CODE GENERATOR
306
REGISTER ALLOCATION
309
CLASS LOADER
308
SECURITY CHECKS
309
CODE VERIFICATION
310
LOGICAL ANALYSIS
311
ADDRESS ASSIGNMENT
312
TREE PARSING
313
MANY OTHER SUB-PROCESSES
314

Fig. 4

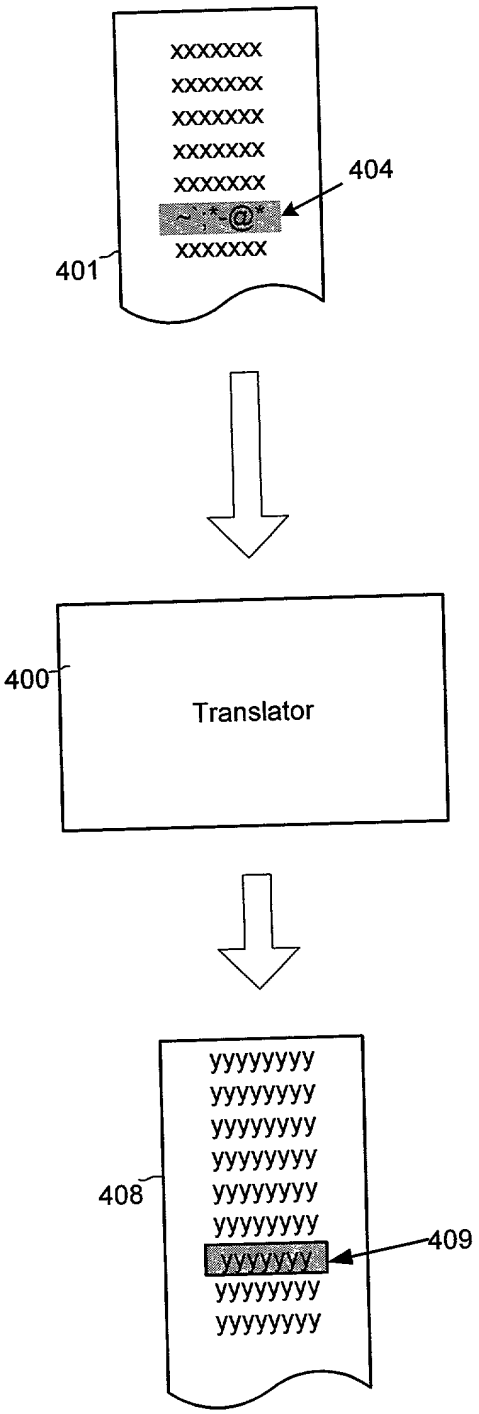

xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
xxxxxxx
~ :*-@*
xxxxxxx
404
401
400
Translator
yyyyyyyy
yyyyyyyy
yyyyyyyy
yyyyyyyy
yyyyyyyy
yyyyyyyy
yyyyyyy
yyyyyyyy
yyyyyyyy
408
409

Fig. 5

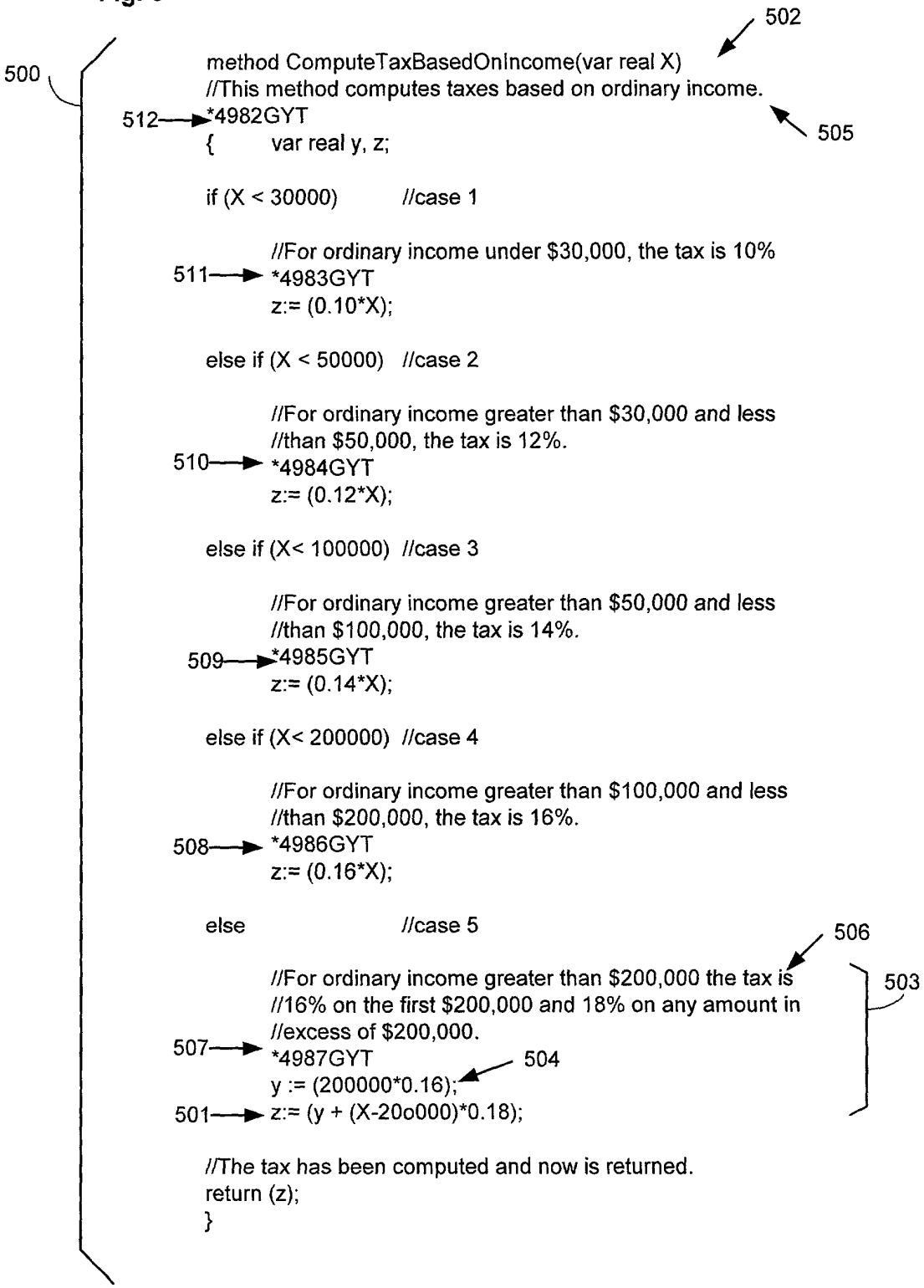

500
502
method ComputeTaxBasedOnIncome(var real X)
//This method computes taxes based on ordinary income.
512
*4982GYT
505
{ var real y, z;
if (X < 30000) //case 1
//For ordinary income under $30,000, the tax is 10%
511
*4983GYT
z:= (0.10*X);
else if (X < 50000) //case 2
//For ordinary income greater than $30,000 and less
//than $50,000, the tax is 12%.
510
*4984GYT
z:= (0.12*X);
else if (X< 100000) //case 3
//For ordinary income greater than $50,000 and less
//than $100,000, the tax is 14%.
509
*4985GYT
z:= (0.14*X);
else if (X< 200000) //case 4
//For ordinary income greater than $100,000 and less
//than $200,000, the tax is 16%.
508
*4986GYT
z:= (0.16*X);
else //case 5
506
//For ordinary income greater than $200,000 the tax is
//16% on the first $200,000 and 18% on any amount in
//excess of $200,000.
503
507
*4987GYT
504
y := (200000*0.16);
501
z:= (y + (X-20o000)*0.18);
//The tax has been computed and now is returned.
return (z);
}

Fig. 6

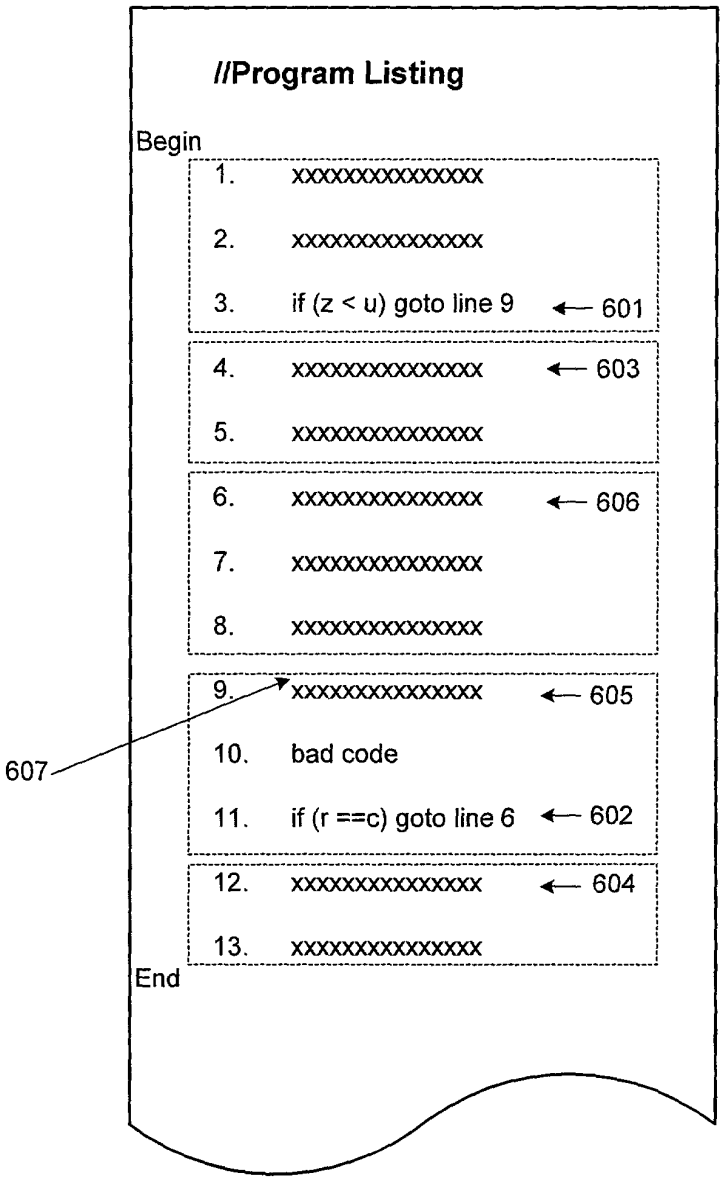

//Program Listing
Begin
1. xxxxxxxxxxxxxxx
2. xxxxxxxxxxxxxxx
3. if (z < u) goto line 9 ← 601
4. xxxxxxxxxxxxxxx ← 603
5. xxxxxxxxxxxxxxx
6. xxxxxxxxxxxxxxx ← 606
7. xxxxxxxxxxxxxxx
8. xxxxxxxxxxxxxxx
9. xxxxxxxxxxxxxxx ← 605
10. bad code
11. if (r ==c) goto line 6 ← 602
12. xxxxxxxxxxxxxxx ← 604
13. xxxxxxxxxxxxxxx
End
607

Fig. 7

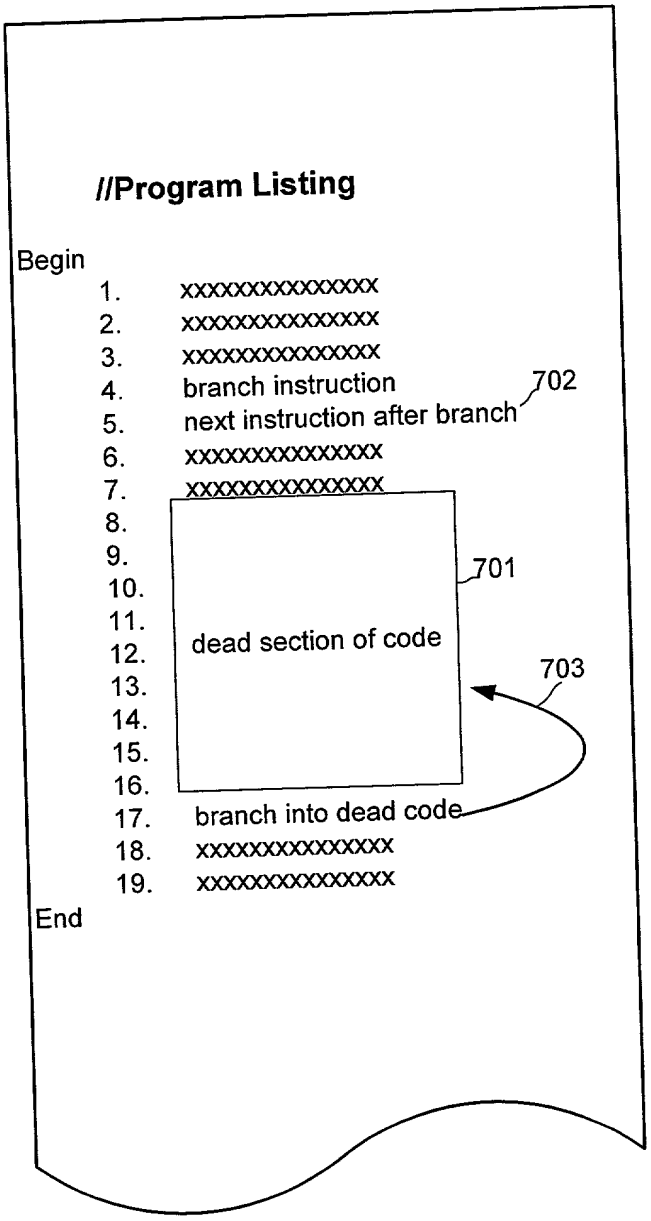
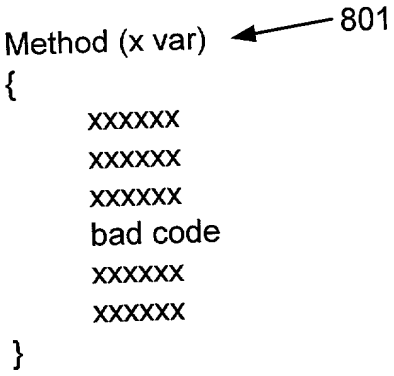

//Program Listing
Begin
1. xxxxxxxxxxxxxxx
2. xxxxxxxxxxxxxxx
3. xxxxxxxxxxxxxxx
4. branch instruction
5. next instruction after branch
6. xxxxxxxxxxxxxxx
7. xxxxxxxxxxxxxxx
8.
9.
10.
11.
12. dead section of code
13.
14.
15.
16.
17. branch into dead code
18. xxxxxxxxxxxxxxx
19. xxxxxxxxxxxxxxx
End
702
701
703

FIG. 8



Docket No. 50277-2000

Fig. 9

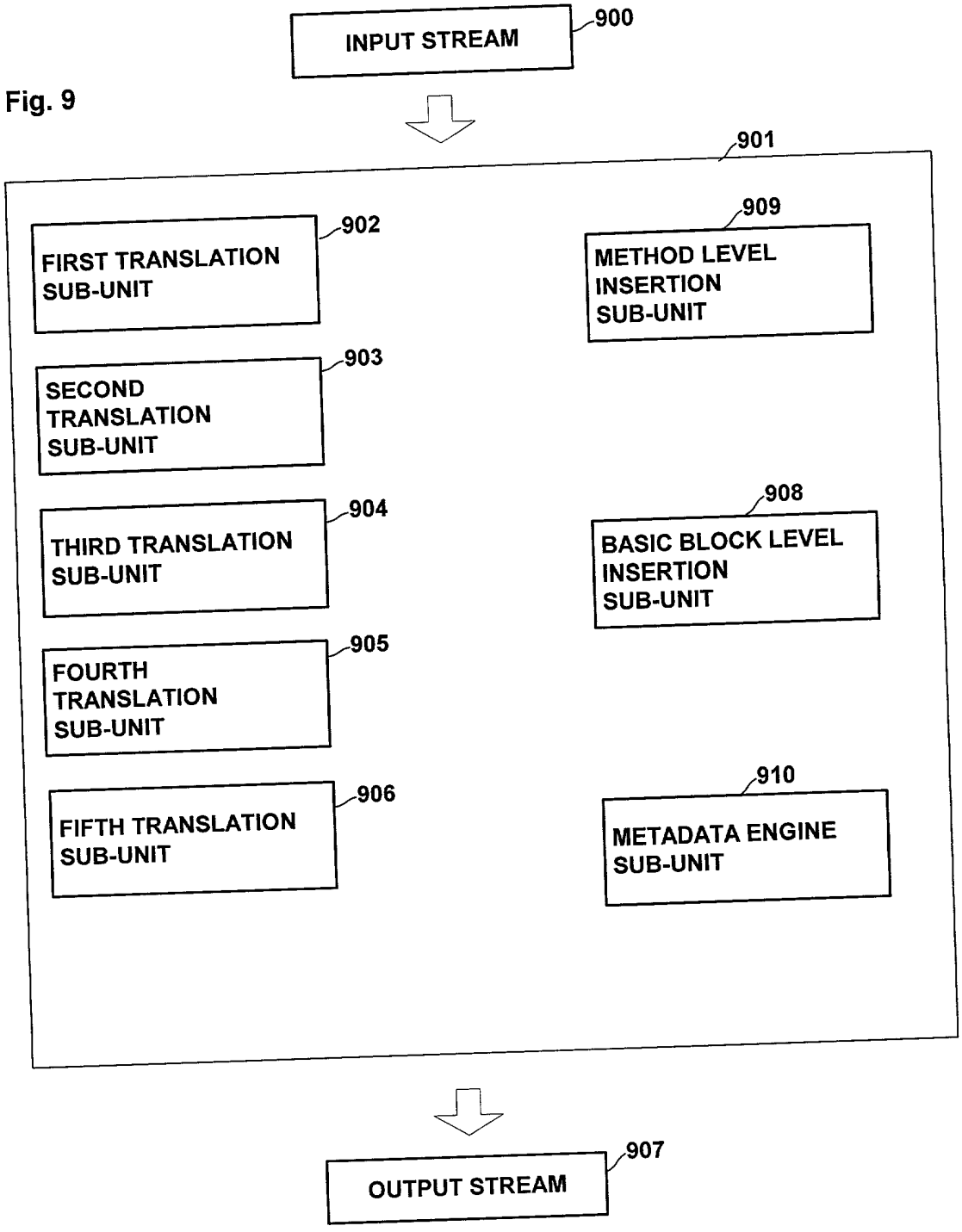

INPUT STREAM
900
901
FIRST TRANSLATION SUB-UNIT
902
SECOND TRANSLATION SUB-UNIT
903
THIRD TRANSLATION SUB-UNIT
904
FOURTH TRANSLATION SUB-UNIT
905
FIFTH TRANSLATION SUB-UNIT
906
METHOD LEVEL INSERTION SUB-UNIT
909
BASIC BLOCK LEVEL INSERTION SUB-UNIT
908
METADATA ENGINE SUB-UNIT
910
OUTPUT STREAM
907

Fig. 10

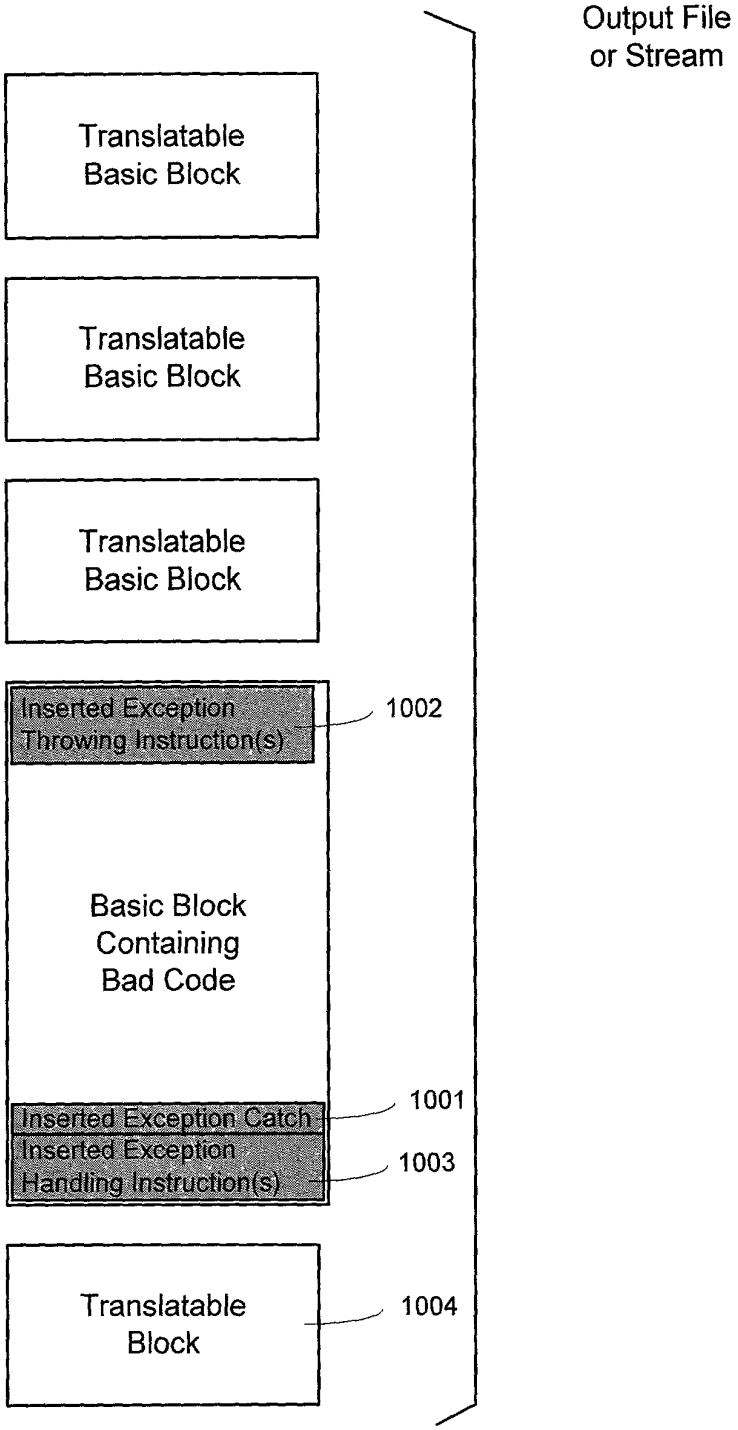

Output File
or Stream
Translatable
Basic Block
Translatable
Basic Block
Translatable
Basic Block
Inserted Exception
Throwing Instruction(s)
1002
Basic Block
Containing
Bad Code
Inserted Exception Catch
1001
Inserted Exception
Handling Instruction(s)
1003
Translatable
Block
1004

Fig. 11

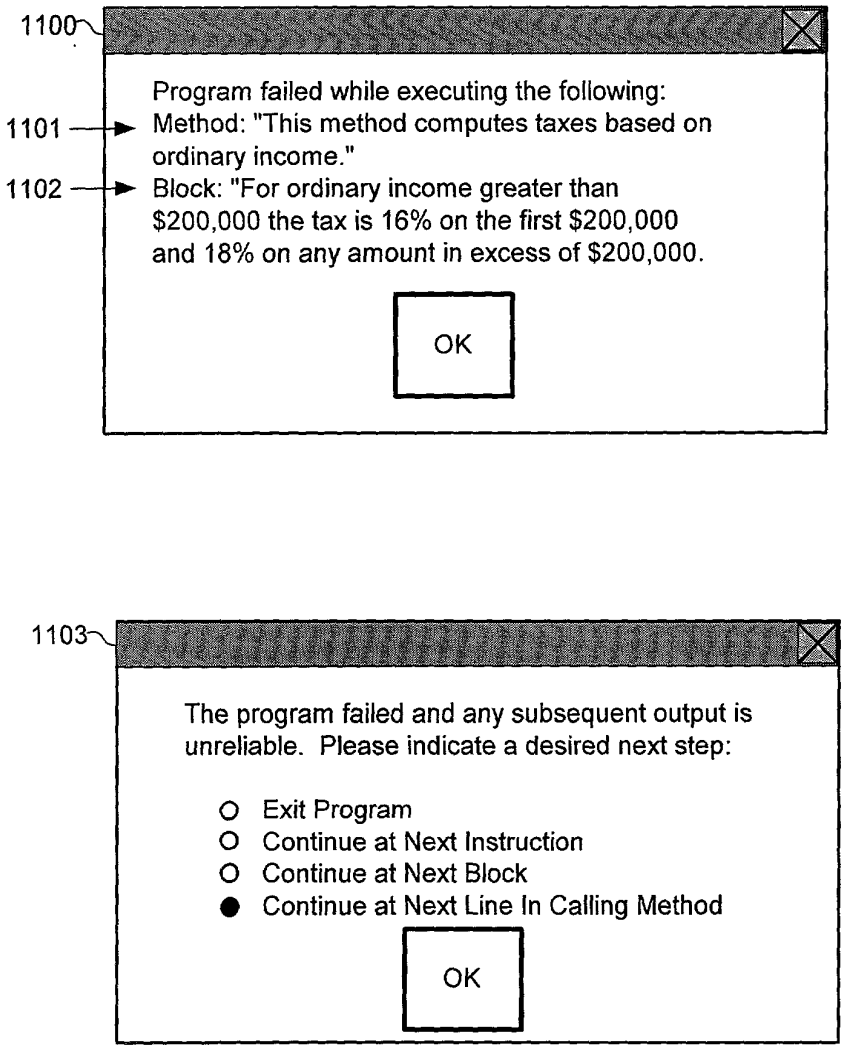
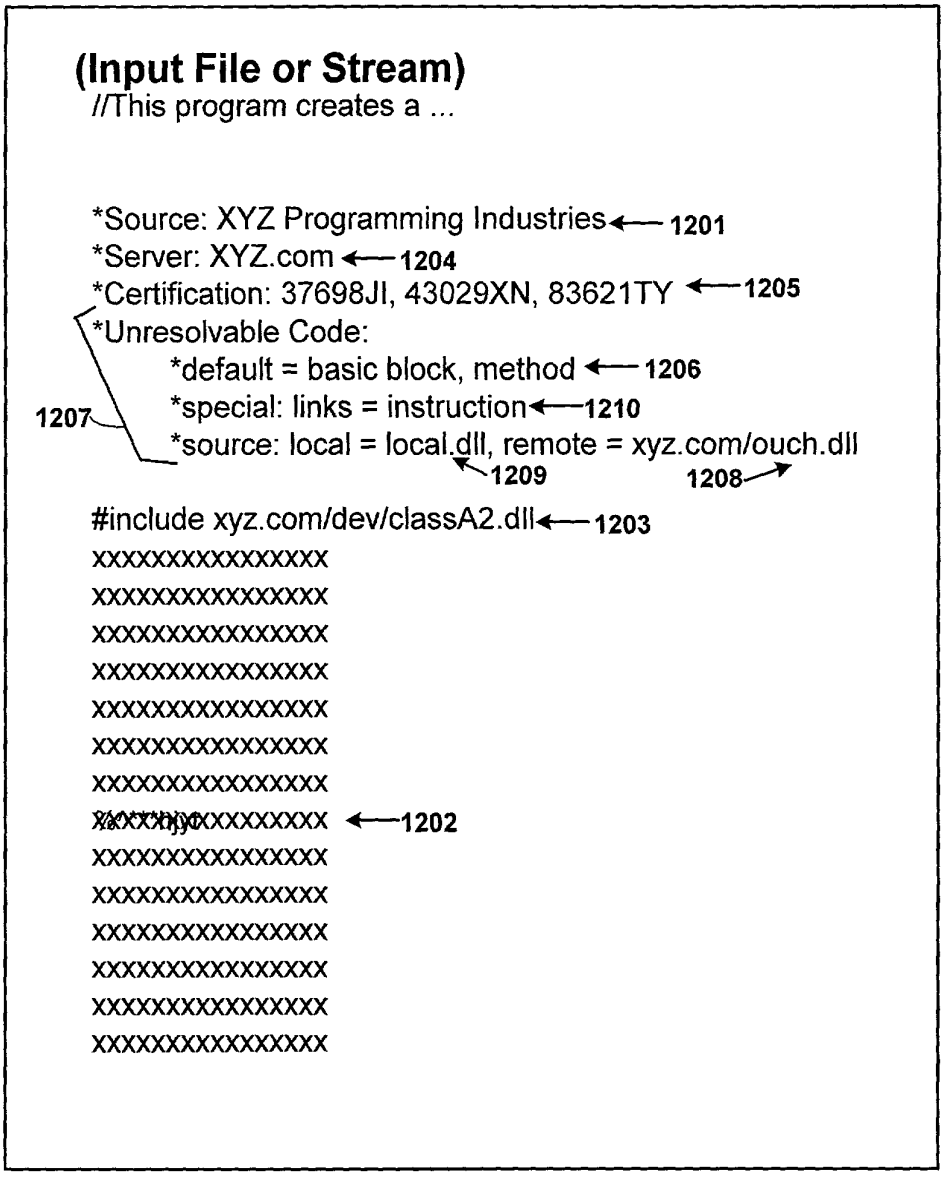

1100
Program failed while executing the following:
1101
Method: "This method computes taxes based on ordinary income."
1102
Block: "For ordinary income greater than $200,000 the tax is 16% on the first $200,000 and 18% on any amount in excess of $200,000.
OK
1103
The program failed and any subsequent output is unreliable. Please indicate a desired next step:
O Exit Program
O Continue at Next Instruction
O Continue at Next Block
● Continue at Next Line In Calling Method
OK

Fig. 12



For all Figures:
'*' = metadata
'//' = comments
'#' = translation time link

Fig. 13

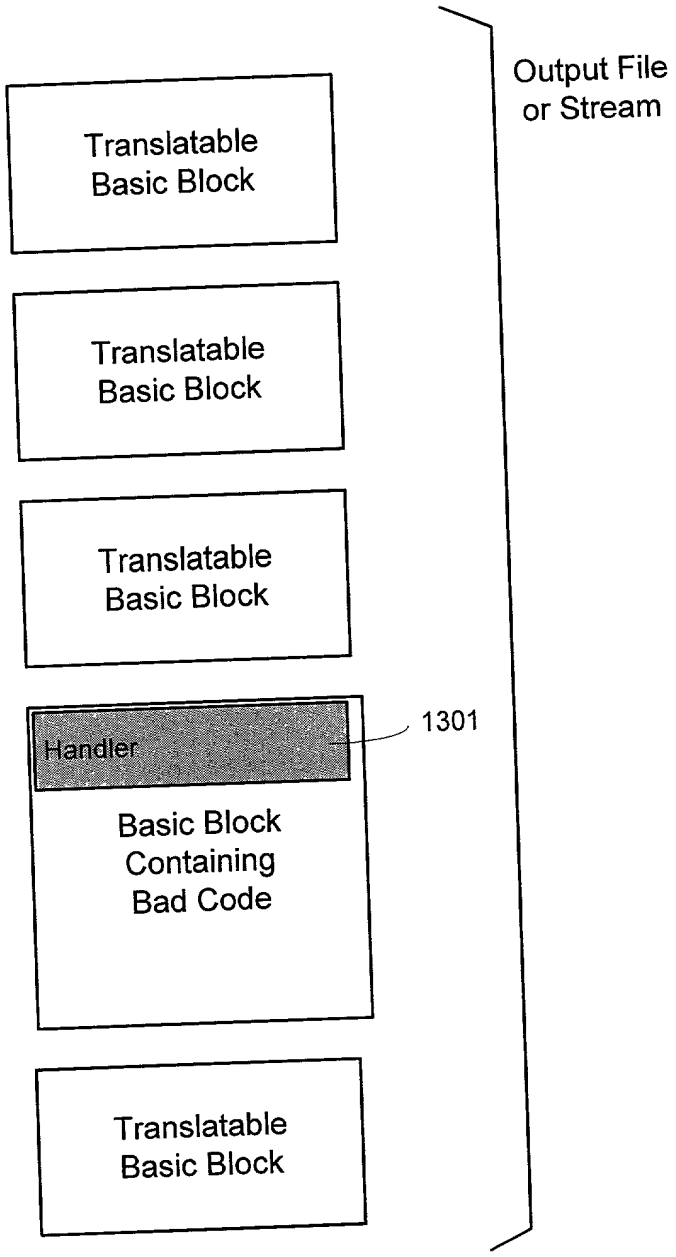

Output File
or Stream
Translatable
Basic Block
Translatable
Basic Block
Translatable
Basic Block
Handler
1301
Basic Block
Containing
Bad Code
Translatable
Basic Block

FIG. 14

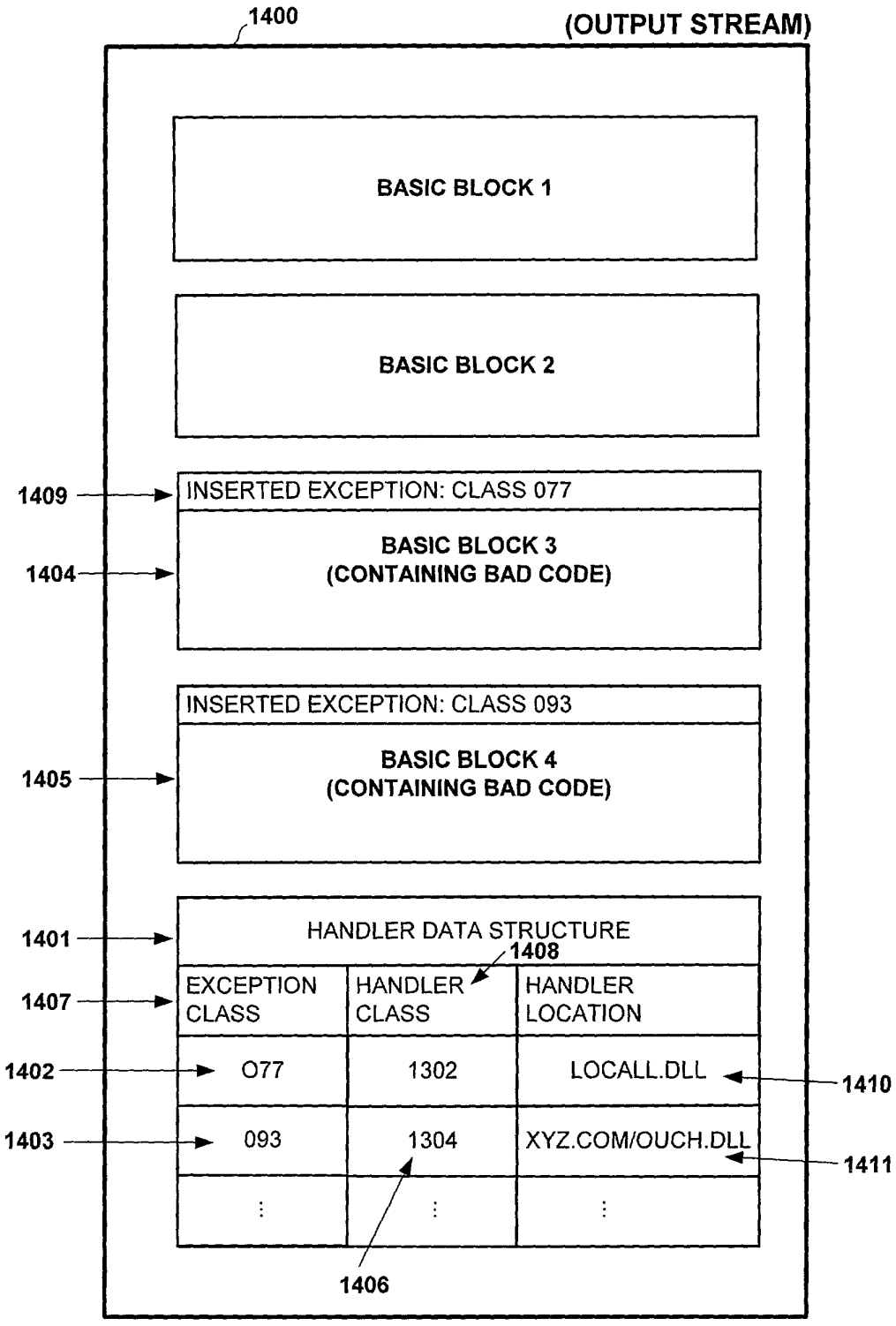

1400
(OUTPUT STREAM)
BASIC BLOCK 1
BASIC BLOCK 2
1409
INSERTED EXCEPTION: CLASS 077
1404
BASIC BLOCK 3
(CONTAINING BAD CODE)
INSERTED EXCEPTION: CLASS 093
1405
BASIC BLOCK 4
(CONTAINING BAD CODE)
1401
HANDLER DATA STRUCTURE
1408
1407
EXCEPTION CLASS
HANDLER CLASS
HANDLER LOCATION
1402
O77
1302
LOCALL.DLL
1410
1403
093
1304
XYZ.COM/OUCH.DLL
1411
1406

Fig. 15

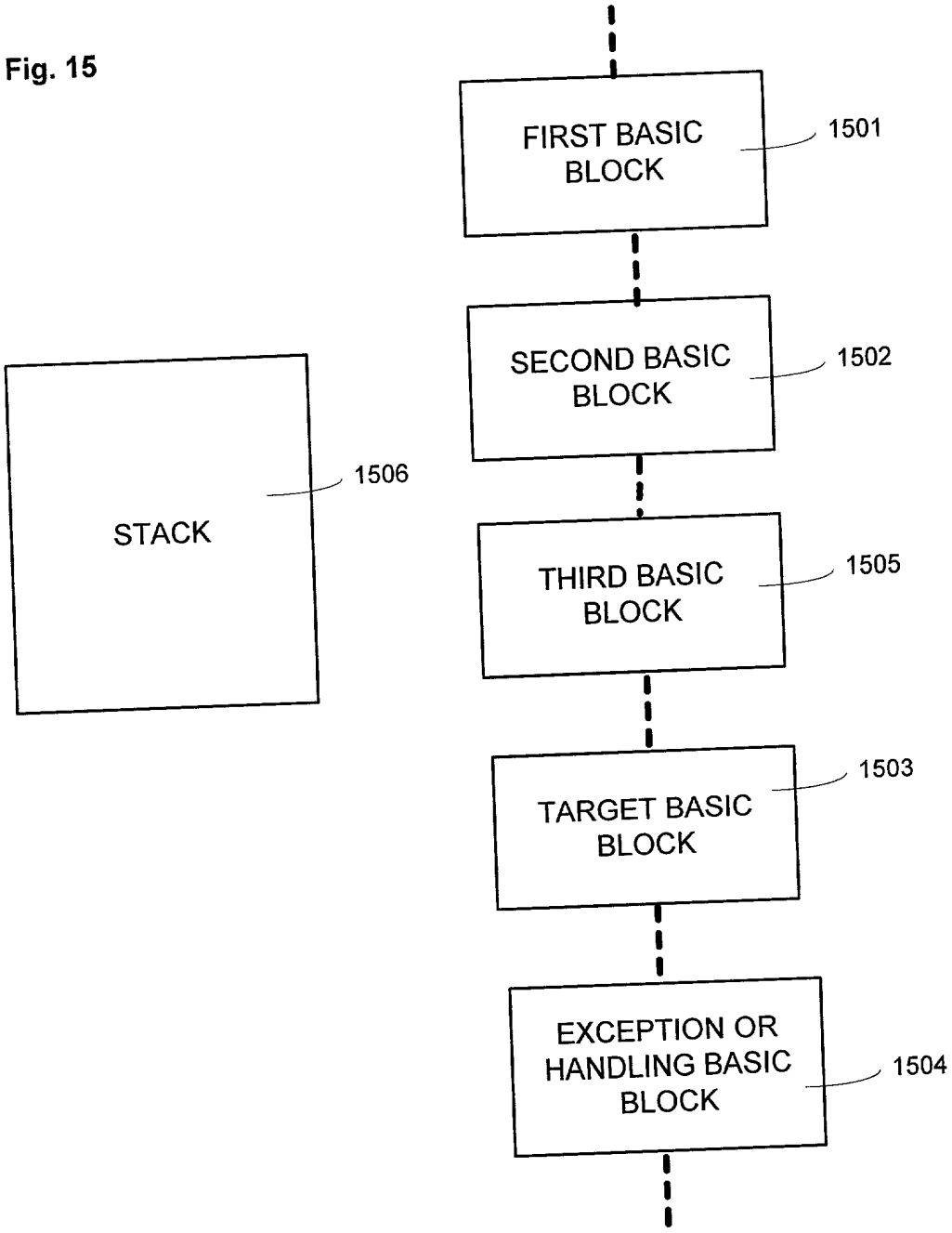

FIRST BASIC BLOCK
1501
SECOND BASIC BLOCK
1502
THIRD BASIC BLOCK
1505
TARGET BASIC BLOCK
1503
EXCEPTION OR HANDLING BASIC BLOCK
1504
STACK
1506

Fig. 16

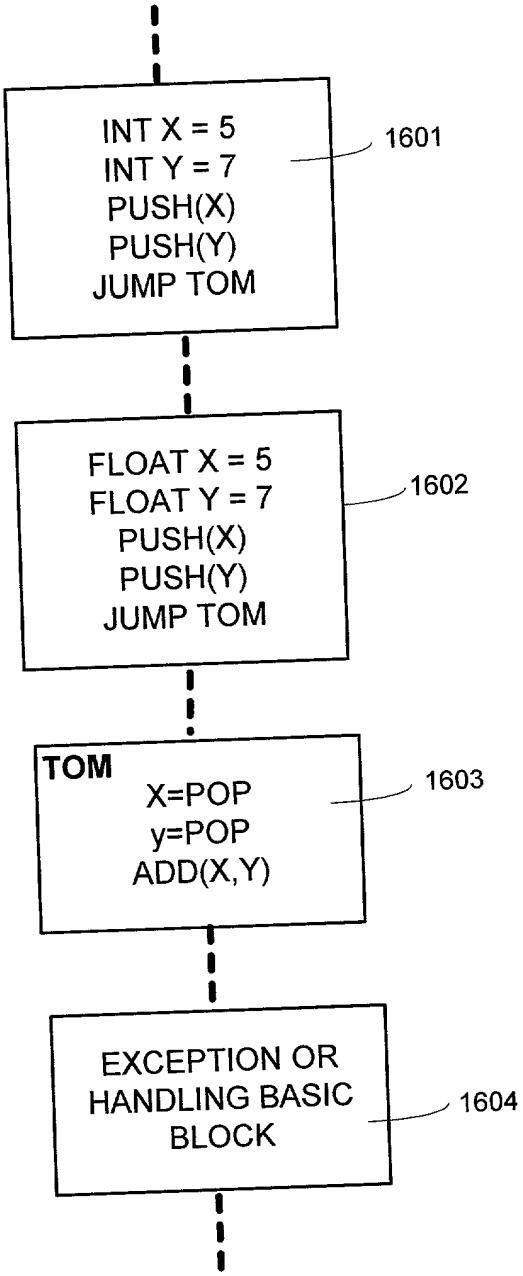
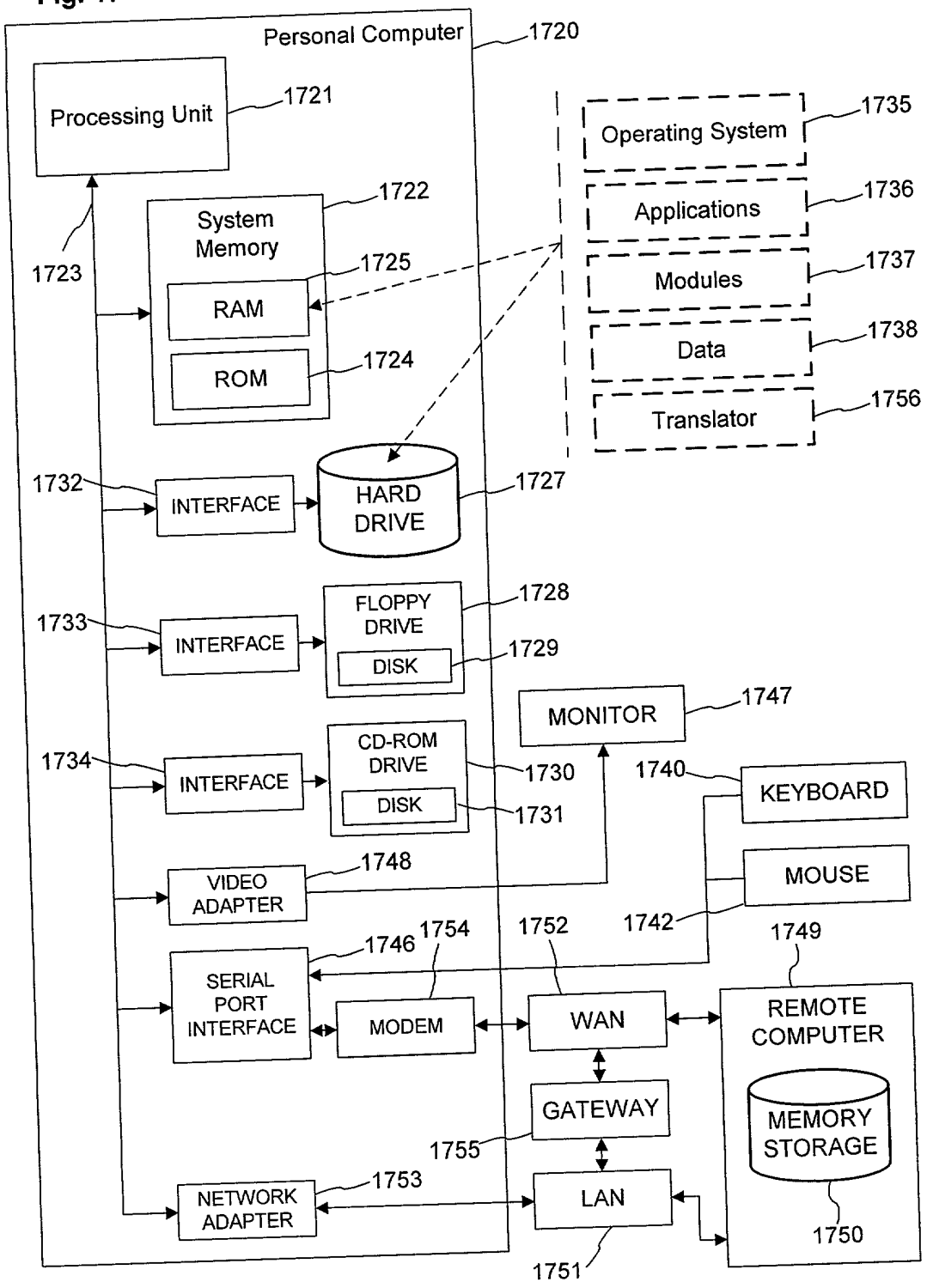

INT X = 5
INT Y = 7
PUSH(X)
PUSH(Y)
JUMP TOM
1601
FLOAT X = 5
FLOAT Y = 7
PUSH(X)
PUSH(Y)
JUMP TOM
1602
TOM
X=POP
y=POP
ADD(X,Y)
1603
EXCEPTION OR
HANDLING BASIC
BLOCK
1604

Fig. 17


Personal Computer
1720
Processing Unit
1721
1723
System Memory
1722
RAM
1725
ROM
1724
Operating System
1735
Applications
1736
Modules
1737
Data
1738
Translator
1756
1732
INTERFACE
HARD DRIVE
1727
1733
INTERFACE
FLOPPY DRIVE
1728
DISK
1729
1734
INTERFACE
CD-ROM DRIVE
1730
DISK
1731
MONITOR
1747
1740
KEYBOARD
MOUSE
1742
VIDEO ADAPTER
1748
SERIAL PORT INTERFACE
1746
1754
MODEM
1752
WAN
1749
REMOTE COMPUTER
MEMORY STORAGE
1750
GATEWAY
1755
NETWORK ADAPTER
1753
LAN
1751